Received 26 October 2025, accepted 23 May 2026, date of publication 30 June 2026
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.25.2.4653

Synthesis of a Multichannel Conflict-free Data-Exchange Device for Mobile Robotic Platforms

IVAN TSMOTS, SOFIIA TESLIUK, YURII OPOTYAK, VOLODYMYR ANTONIV
Lviv Polytechnic National University, Lviv, 79013 Ukraine

Corresponding author: Volodymyr Antoniv (e-mail: volodymyr.y.antoniv@lpnu.ua).

ABSTRACT This article addresses the enhancement of time-division multiple-access (TDMA) for multi-port memory access, the development of a synthesis method for conflict-free multichannel data-exchange devices, and VLSI-oriented component structures. Methods of conflict-free exchange for groups of mobile robotic platforms were analyzed. TDMA was selected for inter-device exchange and was improved by pipelining the memory-access process. The study proposed a multi-port memory controller structure that was targeted at VLSI implementation, interoperates with heterogeneous peripherals and memories, employs a pipelined access path, and synchronizes asynchronously operating devices on the memory bus. A reusable base structure was introduced to shorten synthesis time for parameterized multichannel devices. A configurable clock-generation block was designed to match memory throughput and device count, alongside interface-adaptation blocks were created to bridge transceivers and the controller. A method for synthesizing a multichannel device for conflict-free data exchange has been developed, which ensures the coordination of the intensity of data arrival with the bandwidth of the multiport memory. Modeling of the multiport memory controller on the Cyclone III EP3C16F484C6 FPGA showed that its implementation requires only 114 logic elements and 63 registers. The proposed pipeline of the memory access process allowed to reduce the access cycle time to a value close to the register access time, and the use of a parametric basic structure reduced the complexity of the synthesis of multichannel devices by 58–79%.

KEYWORDS mobile robotic platforms; conflict-free data-exchange methods; multi-port memory; controller; multichannel device; transceiver; interface.

I. INTRODUCTION

Groups of mobile robotic platforms (MRPs) are widely employed across industrial environments, in areas affected by radiological or chemical contamination, and in defense for logistics, reconnaissance, specialized operations, and search-and-rescue tasks. Their motion can be coordinated centrally or in a decentralized manner [1]. Such coordination relies on continuous streams of data from navigation sensors and onboard cameras that describe both each MRP's state and the surrounding environment [1, 2]. Exchange of this information between MRPs and multichannel exchange devices is carried out over a wireless network using transceivers that combine transmitter and receiver functions in a single unit. Multichannel exchange devices must provide parallel, conflict-free, real-time data exchange with a group of MRPs operating at different data rates. These requirements can be satisfied by implementing multichannel data-exchange devices on the basis of multi-port memory (MPM) with time-division allocation of memory resources. At present, the most promising component base for building the memory subsystem is random-access memory (RAM) chips with low access latency and high capacity. In MPM systems, synchronization of access by asynchronously operating external devices to the shared RAM is performed by multi-port memory controllers (MPMCs), whose number corresponds to the number of such devices. External devices are connected to the MPMC via a transceiver–MPMC adaptation block that reconciles the MPMC interface with the transceiver node interface. Configuring multichannel exchange devices to the RAM speed, the speeds of external devices, and their number requires the development of a clock-generation unit (CGU).

Designing MPMCs, transceiver–MPMC adaptation blocks, and the CGU with high technical and operational characteristics demands broad use of modern components (transceivers, microcontrollers, FPGAs) and improvement of the time-division multiple-access (TDMA) method for requests to MPM. Synthesizing, using the MPMC, RAM, adaptation blocks, and the CGU, a multichannel conflict-free data-

exchange device with minimal hardware overhead requires the development of a dedicated synthesis method. The foregoing implies that component design and device synthesis are most effectively carried out using an integrated approach that encompasses the modern component base, the structures of exchange devices and their components, the exchange methods, and the synthesis of exchange devices, all taking into account specific applications and data-arrival intensities.

The study aims to improve time-division multiple access for multi-port memory, devise a synthesis method for conflict-free data-exchange devices, and develop VLSI-oriented components enabling multichannel, conflict-free operation with strong technical and operational characteristics. The objectives are to: (i) enhance TDMA for MPM access; (ii) design the multi-port memory controller; (iii) define a base structure for an MRP-oriented multichannel device; (iv) develop the clock-generation unit; (v) design transceiver-to-MPMC adaptation blocks; (vi) formulate the synthesis method; and (vii) model the MPMC on FPGA and estimate hardware costs.

II. RELATED WORK AND PROBLEM STATEMENT

Analysis of recent studies on the control of groups of MRPs [2, 3] shows that such control relies on collected information about the state of each MRP and the surrounding environment. It is shown in [1] that this information is gathered using a multichannel, conflict-free data-exchange device built on multi-port memory.

The analysis of methods for conflict-free multichannel data exchange within an MRP group indicates that time-division multiple access to memory is predominantly employed, providing collision-free operation, energy efficiency, scalability, minimal latency, and deterministic memory access [3, 4]. The drawbacks of this method include access delays, suboptimal resource utilization, synchronization complexity, and sensitivity to growth in the number of exchanging devices [5, 6]. Examination of time-division multiple access for implementing multichannel conflict-free exchange on an MPM basis [6, 7] shows that such exchange facilitates easy scalability of the number of channels, avoidance of conflicts during parallel exchange, and reduced memory-access latency. A disadvantage of MPM-based multichannel conflict-free exchange using time-division multiple access [7,8] is the difficulty of synchronizing exchanges among asynchronously operating MRPs and the complexity of implementing MPM with conflict-free memory-access mechanisms.

Works [9, 10] analyze the principal MPM structures: memory with independent ports, memory partitioned into independent blocks, dual-port memory, and crossbar-switched memory. The analysis demonstrates that, for fast multichannel conflict-free real-time exchange, it is advisable to choose memory with independent ports, in which each device has its own read-write port and can address all memory cells. Analysis of the independent-port MPM structure [10, 11] shows that its main components are a shared-memory block, controllers connected to a common access bus to the memory block, and clock-generation units.

Based on analysis of independent-port MPM operation, [12, 13] determine that the MPM controller performs the following primary functions: managing access to the shared-memory block, ensuring synchronization and data consistency, and reducing memory-access latency. Analysis of existing MPM controllers [13] shows that they exhibit the following shortcomings: substantial hardware costs to implement memory-access control algorithms, and limited scalability that may require modifying the controller structure or deploying more complex switching schemes when new devices are added.

Analysis of MPM-based multichannel conflict-free exchange devices [15, 16] shows that a clock-generation unit is used to provide time-division of memory access among channels. The analysis in [9] indicates that the main task of such units is to generate clock signals that synchronize read and write operations across different channels. A drawback of existing clock-generation units is their limited ability to adapt to changes in the number of channels and exchange rates.

Studies [17, 18] show that considerable attention in the development of multichannel conflict-free exchange devices is devoted to designing interface-adaptation blocks between the controller and the attached devices. Analysis of such blocks [19] shows that they introduce additional data-transfer latency and incur significant costs associated with the need to support diverse interfaces and communication protocols.

Contemporary research [20] places substantial emphasis on employing modern component bases for implementing multichannel conflict-free data-exchange devices and their components. A review of the literature [21, 22] shows widespread use of modern components (high-speed memory, field-programmable gate arrays (FPGAs), and microcontrollers) alongside modern design tools for such implementations.

Thus, the shortcomings of the existing time-division multiple-access method for MPM and its implementation means are memory-access delay, suboptimal resource utilization, synchronization complexity, and sensitivity to an increasing number of exchanging devices

III. MATERIAL AND METHODS

The following research methods were employed in conducting the study and addressing the stated objectives:

- methods for conflict-free data exchange, the theory of multi-port memory construction, memory-access organization techniques, and real-time systems design theory — to improve the time-division multiple-access method for requests to multi-port memory;
- methods of logical and functional synthesis, the theory of semi-custom and full-custom very-large-scale integration (VLSI) design, and methods of logical and functional modeling — to develop the algorithms and VLSI structures of the components of the multichannel conflict-free data-exchange device (the multi-port memory controller, the interface-adaptation blocks, and the clock-generation unit);
- the theory of digital automata; methods for synthesizing real-time computer systems with coordinated parallel data processing; methods for matching data-arrival intensity to the access intensity of parallel memory; time-division multiplexing techniques; and methods for buffering and synchronizing inter-channel data transfer — to develop a synthesis method for multichannel conflict-free data-exchange devices subject to specified requirements.

The VHDL hardware description language and the Quartus II development environment for Cyclone III EP3C16F484C6 FPGAs were used to model the multi-port memory controller

and to assess the hardware resources required for its implementation.

IV. RESULTS OF SYNTHESIS OF A MULTI-CHANNEL CONFLICT-FREE DATA-EXCHANGE DEVICE FOR MOBILE ROBOTIC PLATFORMS USING MULTI-PORT MEMORY

A. CONFLICT-FREE EXCHANGE METHODS AND AN IMPROVED TIME-DIVISION MULTIPLE-ACCESS SCHEME

Conflict-free data exchange is a method of organizing information transfer among MRPs without conflicts such as collisions, data loss, or corruption. To ensure conflict-free exchange among MRPs, the following methods may be employed [3–6]:

- time-division multiple access (TDMA), where each MRP is assigned a time slot for memory access, thereby avoiding conflicts during write/read operations;
- frequency-division multiple access (FDMA), where each MRP is allocated a specific frequency;
- code-division multiple access (CDMA), which uses dedicated code sequences allowing multiple users to transmit simultaneously within the same frequency band;
- polling, implemented as centralized transmission control, whereby a master MRP sequentially polls subordinate MRPs, granting them the right to transmit.

Among the methods considered, TDMA is the most appropriate for conflict-free exchange among MRPs because it provides collision-free operation, energy efficiency, scalability, minimal latency, and deterministic memory access. Using TDMA together with multi-port memory (MPM) enables simultaneous (write/read) access by multiple MRPs (devices) and improves resource utilization in real-time systems.

In the proposed method, focused on data exchange with mobile robotic platforms, conflicts are eliminated by using the time division multiple access (TDMA) method and multi-port memory, in which each MRP is allocated its own time slot for reading or writing data. Thus, simultaneous access of several platforms to the same resource does not lead to collisions. In an MPM, the principal elements are the memory medium—implemented using random-access memory (RAM)—and multi-port memory controllers (MPMCs) through which external devices are connected. Conflict-free access by external devices to RAM is achieved via TDMA: each device is cyclically assigned a fixed time interval for RAM access determined by the RAM access-cycle time. The period with which devices are cyclically granted access to RAM is set by the access time of the fastest external device. Knowing the smallest period at which devices can access the MPM and the RAM throughput, one can determine the number of devices that can access the MPM without conflict. This number is given by the following formula:

$$m = \left\lfloor \frac{T_{min}}{t_{c\ RAM}} \right\rfloor, \quad (1)$$

where T_{min} is the smallest access period of the external devices to the MPM; t_{cRAM} is the RAM access-cycle time; and m is the number of external devices with conflict-free access to RAM.

To reduce the (read-write) access time to the MPM, it is proposed to pipeline the write (read) process from an external device to RAM by dividing it into two sub-processes: the first preparatory, and the second the direct RAM access.

Write or read operations in the MPM begin with the preparatory stage, in which the address is written to the address counter. In the second stage, data are read from RAM at the given address and written to buffer memory (pipeline registers).

In write mode, during the preparatory cycle the input data are written to buffer memory (pipeline registers). In the second stage of write mode, the input data are written to RAM and the address counter is incremented by one.

In read mode, during the preparatory stage data are read from buffer memory (pipeline registers) and the address counter is incremented by one. In the second stage of read mode, data are read from RAM and written to buffer memory (pipeline registers).

To ensure conflict-free access by external devices to the MPM, a cyclic sequence of clock pulses TI_{1j}, TI_{2j} is formed. An example of generating the cyclic sequence TI_{1j}, TI_{2j} for four external devices (m = 4) is shown in Fig. 1.

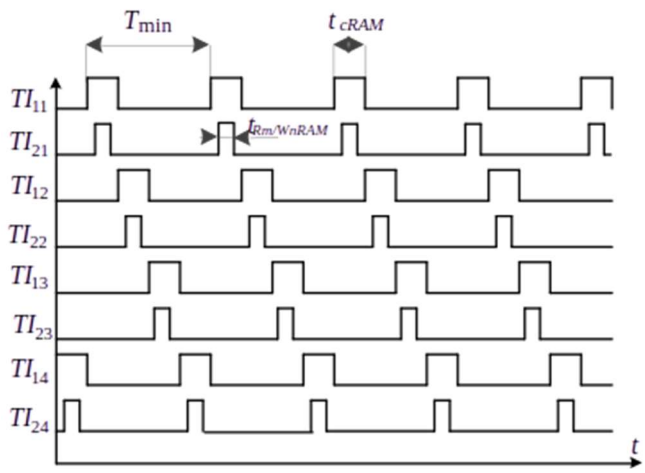


Figure 1. Timing diagram for generating clock pulses for four devices

The durations of the clock TI_{1j}, TI_{2j} are equal to the RAM access-cycle time and the RAM write (read) time, respectively. The minimum period of access to the MPM can be increased by performing serial-to-parallel conversion of input data and parallel-to-serial conversion of data read from RAM. Using such conversions requires increasing the RAM word width.

B. DEVELOPMENT OF THE MPMC STRUCTURE

The structure of the MPMC must be oriented toward implementation as a very-large-scale integration (VLSI) design. The number of MPMCs should match the number of external devices used for data exchange. All MPMCs must be identical and provide the following capabilities:

- operation with RAMs and external devices of differing speeds;
- parallel instantiation of the required number of MPMCs;
- synchronization of RAM access for asynchronously operating external devices;
- pipelining of RAM write and read processes;
- decomposition of the RAM-access process into two sub-processes: the first preparatory, and the second the direct RAM access;

- assignment to each MPMC of periodic, fixed time intervals for RAM access.

An MPMC structure that satisfies these requirements is shown in Fig. 2, where *F1, F2,* and *F3* are nodes for latching input signals (and generating output control signals), Lich*Ad* is the address counter, and *Rg1* and *Rg2* are buffer registers.

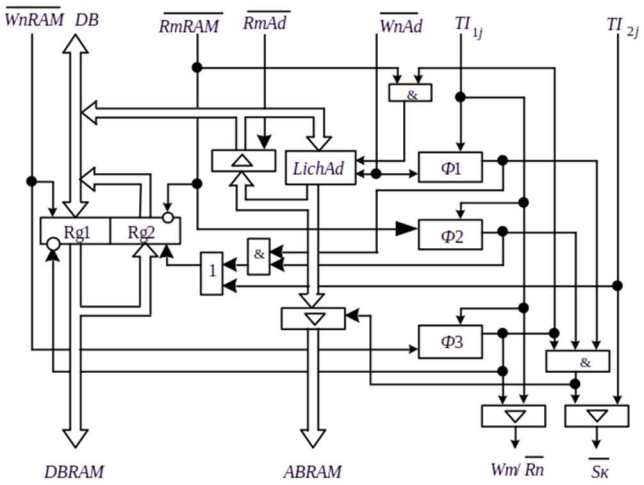


Figure 2. Structure of the MPMC.

The principal components of the MPMC are the input-signal latching and output control-signal generation nodes *F1, F2*, and *F3*, the address counter Lich*Ad*; buffer registers *Rg1* and *Rg2*, and bus drivers [23]. The buffer registers *Rg1* and *Rg2* enable a pipelined RAM-access mode, in which the access process is divided into two cycles: the first preparatory, the second the direct RAM access. The nodes *F1, F2*, and *F3* are of the same type and are intended to synchronize RAM access by asynchronously operating devices by latching the input control signals $\overline{WnRAM}$, $\overline{RmRAM}$, $\overline{WnAd}$ and generating output control signals (negative-polarity single pulses).

Node for latching an input signal and generating an output control signal. A circuit for latching the input signal and generating the output control signal has been developed, it is shown in Fig. 3.

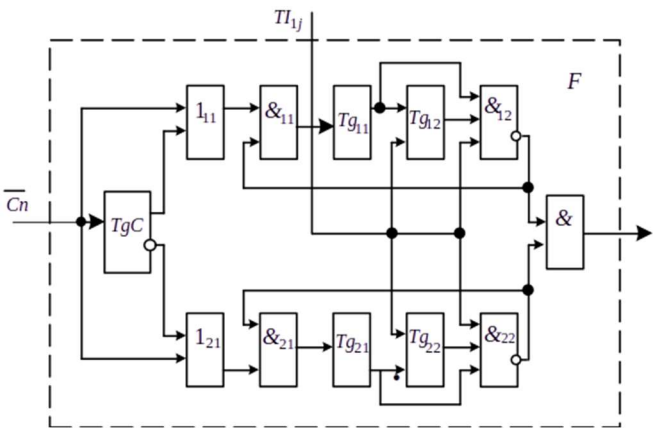


Figure 3. Circuit of the input-signal latching and output control-signal generation node.

The node operates as follows. Before operation begins, triggers Tg_{11}, Tg_{12}, Tg_{21}, and Tg_{22} are in logic "0" On the rising edge (transition from logic "0" to logic "1") of the control input signal $\overline{Cn}$, either trigger Tg_{11} (when trigger *TgC* is at logic "0") or trigger Tg_{21} (when trigger *TgC* is at logic "1") is set to "1" and the state of *TgC* toggles. A logic "1" at the output of Tg_{11} (or Tg_{21}) is applied to the data input of Tg_{12} (or Tg_{22}) and, on the rising edge of the clock pulse TI_{1j}, is latched into that trigger. When logic "1" levels are present at the second and third inputs of the element $NAND_{12}$ (or $NAND_{22}$) and a TI_{1j} clock pulse arrives, a negative pulse is formed at the output of that logic element. This pulse appears at the node output *F* and also passes through the element AND_{11} (or AND_{21}), whose rising edge resets trigger Tg_{11} (or Tg_{21}) to logic "0" Upon the arrival of subsequent control signals $\overline{Cn}$, the latching and control-signal generation node *F* operates analogously.

The developed structure of a multiport memory controller for multichannel conflict-free data exchange combines pipelined read and write processes, hardware synchronization of asynchronous requests from external devices, and time-sharing of memory access. This combination provides reduced access time, scalability of the number of channels, and matching the intensity of data flows with memory bandwidth in real-time systems.

C. DEVELOPMENT OF A BASELINE STRUCTURE FOR A MULTICHANNEL CONFLICT-FREE DATA-EXCHANGE DEVICE FOR MRPS

To develop a multichannel conflict-free data-exchange device for MRPs, the improved time-division multiple-access method for accessing MPM is adopted. The device structure is determined primarily by the number of MRPs connected to the MPM. In addition, a computer system (CS) is connected to the MPM to process data and, on this basis, control the MRP group. The principal components used to synthesize the multichannel conflict-free device are: MPMCs, transceivers, transceiver–MPMC adaptation blocks, CS–MPMC adaptation blocks, and the clock-generation unit (CGU).

In the multichannel device, the MPMCs synchronize access to the MPM for asynchronously operating MRPs and the CS. Data exchange between MRPs and the device is carried out over a wireless network using transceivers. To reconcile the MPMC interface with the transceiver and CS interfaces, transceiver–MPMC and CS–MPMC adaptation blocks are employed. The CGU is used to configure the device for the number of external devices and the RAM throughput.

The RAM capacity depends on both the number *m* of devices connected to the MPM and the sizes N_j of the data arrays exchanged by the devices. In addition, the multichannel device must provide a memory volume *V* for storing data used to control the exchange process. Based on these considerations, the RAM capacity is determined as:

$$Q = V \times n_{RAM} + \sum_{j=1}^{m} N_j \times n_{RAM}, \qquad (2)$$

where n_{RAM} is the word width.

The most promising component base for the memory medium is RAM chips with a short access-cycle time t_{cRAM} and large capacity *Q*.

Data exchange among MRPs is performed via a wireless network using transceivers that combine transmitter and receiver functions in a single device, enabling both transmission and reception on one antenna. The main components of a transceiver are:

- receiver path, including the antenna, filters, amplifiers, and frequency converters;
- transmitter path, including the power amplifier,

modulator, filters, and other components preparing the signal for radiation via the antenna;

- antenna, serving as the interface to the wireless medium, transmitting and receiving radio signals;
- amplifiers and filters, used to amplify signals and suppress noise to improve signal quality during transmission and reception;
- microcontroller, which controls transceiver operation, including data processing and transmit/receive management;
- modulator and demodulator, which convert the signal to a form suitable for transmission over the channel and back to its original form upon reception;
- frequency converters, used to shift the signal frequency for transmission or reception;
- external interfaces, which provide interconnection with other devices or systems.

The developed baseline structure of the multichannel data-exchange device using MPM is shown in Fig. 4, where CS denotes the computer system; CGU the clock-generation unit; MPMC the multi-port memory controller; RAM the random-access memory; DB the data bus; AB the address bus; $Rm/\overline{Wn}$ the read-write signal; and $\overline{Sk}$ the chip-select signal.

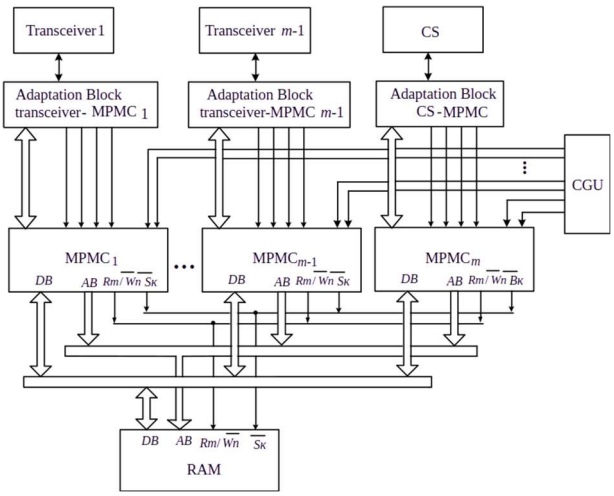


Figure 4. Baseline structure of the multichannel data-exchange device using MPM.

In the multichannel device for MRPs, the intensity of access to the MPM with time-division of resources depends on the RAM access time as well as on the time to form control signals, addresses, and data. A short formation time for control signals, address generation, and data delivery to the RAM inputs can be ensured by implementing these functions in hardware within the MPMCs.

In the multichannel device, read or write transfers to RAM with time-division of resources can be performed sequentially or in arrays. Working with data arrays requires that the address be written to the corresponding MPMC's address counter Lich*Ad* once at the start of the array access. Under sequential access by external devices to the MPM, each write or read requires a prior address write.

D. DEVELOPMENT OF THE STRUCTURE OF THE CLOCK-GENERATION UNIT

The CGU must generate continuous sequences of clock pulses TI_{1j} and TI_{2j}, where $j=1,..,m$, whose durations are equal, respectively, to the RAM access-cycle time and the RAM write (read) time. A CGU structure that generates these clocks is shown in Fig. 5, where *TI* is the clock input; *ROM* is read-only memory; Lich*TI* is the clock-pulse counter; *Rg* is a register; *Tg* is a trigger; *SP* is a comparator; *Ad* is the address input; TI_{11}, $TI_{21}, \ldots, TI_{1m}$, TI_{2m} are the outputs providing the clock-pulse sequences for 1,..,m devices; *PU* is the initial set; *Ad* (upper address bits) is the input of the higher-order address lines; M_j is the input specifying the number of clock pulses; and *Wp* is the write-control input.

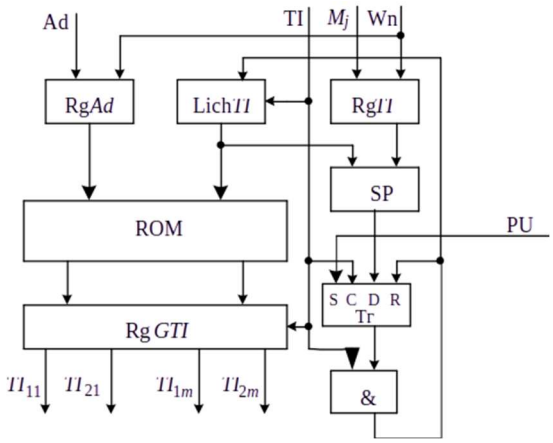


Figure 5. Structure of the CGU

CGU operating procedure. Before operation, the ROM is programmed with data that produce the sequences TI_{1j} and TI_{2j} for different numbers of external devices. The CGU is then configured by performing the following actions:

- write, with a positive pulse at *Wp*, the values from inputs *Ad* and M_j into registers Rg*Ad* and Rg*TI*, respectively;
- set the trigger *Tg* to logic "1" with a positive pulse at PU.

The data $Ad = \lceil log_2 m \rceil$ (where $\lceil \quad \rceil$ sign of a greater whole; *m* is the number of devices connected to the MPM) from Rg*Ad* are applied to the upper address bits of the ROM to select the memory region that stores the clock-generation program for *m* devices. The register Rg*TI* holds the value $M_j=3m$, which specifies the number of *TI* clock pulses in one period of generating $TI_{11}, \ldots, TI_{1m}$ and $TI_{21}, \ldots, TI_{2m}$. In CGU adaptation to RAMs with different speeds is achieved by changing the frequency of the input clock TI.

At start-up, the AND gate output produces a positive pulse upon the arrival of a positive TI edge, which sets *Tg* and clears Lich*TI* to zero. On the falling edge of TI (transition from logic "1" to "0"), data are read from the ROM and written into Rg*GTI*. On the first rising edge of TI (transition from logic "0" to "1"), the Lich*TI* counter increments by one. Data are fetched from the ROM at the corresponding address, on the subsequent falling edge, are written into Rg*TI*. With each subsequent TI pulse, Lich*TI* increments by one, ROM data are read, and those data are written into Rg*GTI*. When the counter state equals *3m* (the value stored in Rg*TI*), the comparator SP asserts logic "1," which is latched into Tg on the falling edge of TI. Upon the next positive TI pulse, the AND gate again produces a positive pulse that sets Tg and clears Lich*TI* to zero.

With subsequent clock pulses, the CGU operates analogously. The outputs TI_{11}, $TI_{21}, \ldots, TI_{1m}$, TI_{2m} generate the required sequences of clock pulses, which are applied to the clock inputs of the corresponding MPMCs.

E. DEVELOPMENT OF ADAPTATION BLOCKS TO THE MPMC

To implement the communication channel and the corresponding transceiver block, the Silicon Labs Si4463 IC was selected, operating in the 433 MHz band, together with the industrial HC-12 module built on this chip. The transceiver supports multiple channels configured by commands with a 400 kHz step (up to 100 channels total). The module's maximum transmit power is 100 mW (20 dBm); receiver sensitivity is −117 dBm at an over-the-air rate of 5,000 baud. The link distance is 1,000 m (mode FU3 at 4,800 bps on the UART) and up to 1,800 m in mode FU4 at a reduced data rate and payload. Notably, the module integrates a microcontroller that manages the transceiver, eliminating the need for user programming.

The module provides transparent half-duplex serial data transfer via a UART. It supports transparent transmission modes that the user selects using AT commands. The typical standby currents for modes FU1, FU2, FU3, and FU4 are 3.6 mA, 80 µA, 16 mA, and 16 mA, respectively, while the maximum operating current in any mode is 100 mA (during transmission).

The Si4463 communicates over SPI with an STM8S003F3 microcontroller (STMicroelectronics). Using the STM8S provides a transparent serial data interface, allowing the HC-12 to be used like a cable-replacement TTL serial interface without additional hardware. The configured UART and receiver settings are stored in non-volatile flash memory. Device configuration and control are performed over SPI, leveraging both the built-in packet handler and a 64-byte FIFO buffer, as well as direct GPIO control. Owing to their high level of integration, the transceivers require a minimal number of external components.

A structural scheme for connecting the transceiver to the MPMC is shown in Fig. 6, where SPI is the 4-wire serial interface and I²C is the serial communication interface.

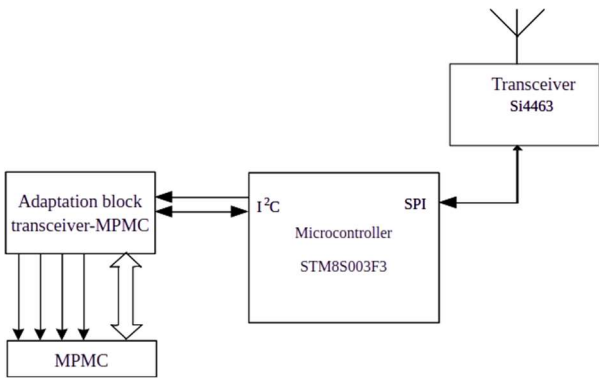


Figure 6. Structural scheme for connecting the transceiver to the MPMC

The transceiver–MPMC adaptation block must reconcile the I²C interface with the MPMC interface. Serial-to-parallel and parallel-to-serial conversions are used, respectively, for writes to and reads from RAM.

The structure of the transceiver–MPMC adaptation block is shown in Fig. 7, where CGND is the control-signal generation node; Rg is a register; the control signals $\overline{WnAd}$, $\overline{Rm\ Ad}$, $\overline{RmRAM}$ and $\overline{WnRAM}$ denote address write, address read, RAM read, and RAM write, respectively; DB is the data bus; and I²C is the serial interface.

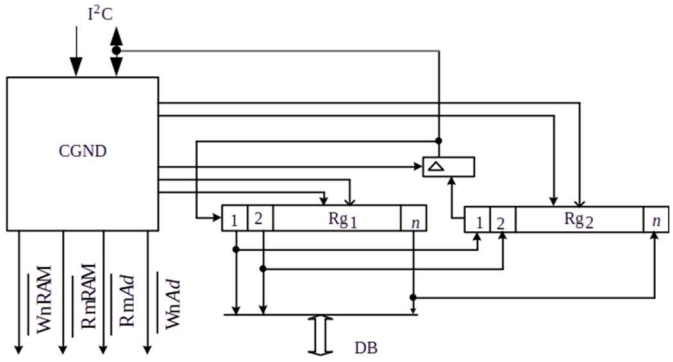


Figure 7. Structure of the transceiver–MPMC adaptation block

The conversions in the adaptation block satisfy the following relation:

$$n_{in}F_{in} = n_{out}F_{out}, \tag{3}$$

where n_{in} is the input word width, F_{in} is the input data-arrival frequency, n_{out} is the output word width, and F_{out} is the output data frequency. Serial-to-parallel conversion increases n_{out} and correspondingly reduces F_{out} at the MPMC input. In this case,

$$F_{out} = \frac{n_{in}F_{in}}{n_{out}}. \tag{4}$$

Reducing F_{out} relaxes RAM speed requirements and increases the number of devices that can be connected to the MPM. To ensure uniformity of adaptation blocks, the CS is also connected to the MPMC via the I²C interface.

F. SYNTHESIS METHOD FOR A MULTICHANNEL CONFLICT-FREE DATA-EXCHANGE DEVICE

The input information for synthesizing a real-time multichannel conflict-free data-exchange device comprises:

- the baseline structure of the multichannel conflict-free device;
- the developed structures of the MPMC, the adaptation blocks, and the CGU;
- the number of external devices (MRPs and the CS) connected to the device;
- the MPMC interfaces;
- the transceiver type and its communication interfaces;
- the microcontroller type used to control the transceiver;
- the interfaces between the microcontroller, the transceiver, and the MPMC;
- the interface between the CS and the MPMC;
- the transceiver–MPMC and CS–MPMC adaptation blocks;
- the CGU, configurable for the number of devices and the data-arrival frequency, for generating the clock sequences TI_{1j} and TI_{2j};
- the transceiver data-arrival frequency F_d and word width n_d;
- the word width and array sizes N_j used in exchanges with MRPs;
- the memory volume V for control data;
- tech-economic requirements and constraints.

For conflict-free real-time data exchange, the device must sustain data arriving with intensity:

$$P_d = mn_d F_d. \quad (5)$$

The synthesis problem for a real-time multichannel device based on MPM is thus to ensure conflict-free exchange at arrival intensity P_d with minimal hardware cost.

Minimal hardware cost is achieved by matching the data-arrival intensity P_d to the MPM service intensity:

$$P_{MPM} = mn_{RAM} F_{MPM}, \quad (6)$$

where n_{RAM} is the RAM word width and F_{maxMPM} is the maximum data-accept rate at the MPM inputs. Matching P_d and P_{MPM} can be accomplished by adjusting the maximum input data rate to the MPM and by reducing the MPM access-cycle time.

In the multichannel device, serial-to-parallel or parallel-to-serial conversion in the adaptation blocks can change the data-arrival frequency at the MPM input. Each conversion is characterized by the coefficient

$$\beta = \frac{n_d}{n_{RAM}}, \quad (7)$$

where n_d is the data width at the adaptation-block inputs and n_{RAM} is the RAM word width. The value of β determines the conversion type: β <1 (serial-to-parallel), β >1 (parallel-to-serial). Using serial-to-parallel conversion allows reduction of the maximum input frequency to the MPM:

$$F_{max\,MPM} = \beta F_d. \quad (8)$$

The value of the coefficient β during concatenative-parallel conversion determines the words widths that are written to or read from RAM. Which also determines the widths of the pipeline registers in the MPMC.

Given the number of external devices connected to the multichannel device and the maximum input frequency at the MPM, the MPM access-cycle time is

$$t_{cMPM} = mF_{MPM}, \quad (9)$$

where t_{cMPM} is the MPM access-cycle time.

Access to the MPM uses time-division multiple access with a pipelined memory-access process. Pipelining reduces the access-cycle time so that

$$t_{cMPM} = t_{Rm/Wn\ RAM} + t_{Rg}, \quad (10)$$

where t_{Rg} is the register-access time and $t_{Rm/WnRAM}$ is the RAM read-write time. The parameter $t_{Rm/WnRAM}$ is used to select RAM devices that meet the required speed.

G. MODELING AND HARDWARE-COST EVALUATION OF THE MPMC IMPLEMENTATION ON FPGA

The MPMC was modeled in the VHDL hardware description language using the Quartus II development environment on an FPGA of the Cyclone III family EP3C16F484C6. The principal components of MPMC$_j$ are the nodes *F1, F2, F3*; the address counter Lich*Ad*; and a tri-state parallel register.

The nodes *F1, F2,* and *F3* are identical and are intended to latch the control signals $\overline{WnAd}$, $\overline{RmRAM}$, $\overline{WnRAM}$ and, within the allocated time interval generate TI_{1j}, and at their outputs single negative pulses. In the MPMC, *F1, F2,* and *F3* nodes ensure synchronization of RAM access by asynchronously operating external devices. The symbolic notation of the node for latching an input signal and forming an output control signal *F* used in the model is shown in Fig. 8.

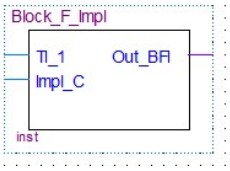


Figure 8. Symbolic notation of the F node for input-signal latching and output control-signal generation.

The schematic of the *F* node for input-signal latching and output control-signal generation is shown in Fig. 9.

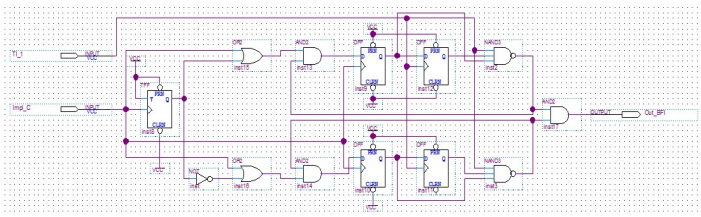


Figure 9. Schematic of the F node for input-signal latching and output control-signal generation

Operation of node *F*. When one of the control signals $\overline{WnAd}$, $\overline{RmRAM}$, $\overline{WnRAM}$ arrives at input *Impl_C*, it is latched in triggers inst9 or inst10. Upon the arrival of a positive clock pulse TI_{1j} (the allocated fixed time slot) at input TI_1 the node produces at output Out_BFI a negative-polarity pulse. The timing diagram of node F is shown in Fig. 10.

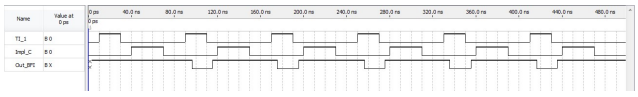
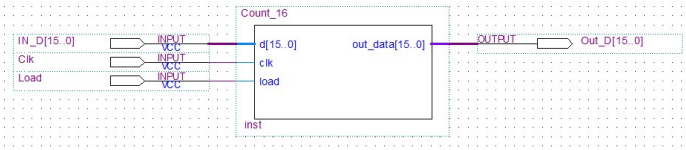


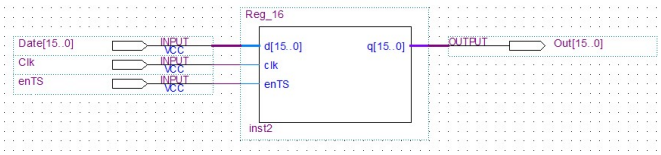
Figure 10. Timing diagram of the F node for input-signal latching and output control-signal generation

From the timing diagram it follows that the output Out_BFI produces a negative pulse whose duration equals the duration of the positive clock pulse TI_{1j}.

For the MPMC model, symbolic notations of a 16-bit address counter and a 16-bit parallel tri-state register are used, shown in Figs. 11a and 11b, respectively.



a)



(b)

Figure 11. Symbolic notation: (a) 16-bit address counter; (b) 16-bit parallel register with tri-state output

The widths of both the address counter and the tri-state register are 16 bits. In the symbolic notation of the address counter, the ports are assigned as follows: IN_D[15..0] – 16-bit input data loaded into the counter; Clk – clock pulses whose rising edge increments the counter by one; Load – load signal for writing input data into the counter; Out_D[15..0] – 16-bit address output.

In the symbolic notation of the parallel register with tri-state outputs, the ports are assigned as follows: Data[15..0] – 16-bit input data written to the register; Clk – input clock whose rising edge writes the input data to the register; enTS – signal that places the register output in the tri-state; Out[15..0] – 16-bit output. The symbolic notation of the MPMC is shown in Fig. 12.

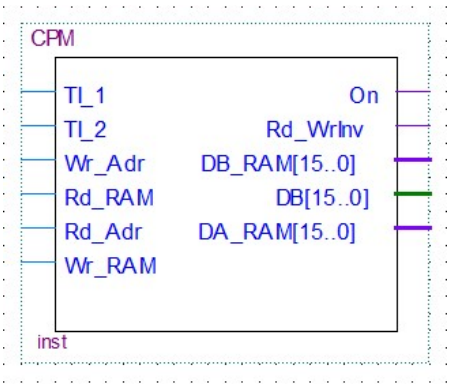


Figure 12. Symbolic notation of the MPMC.

In the symbolic notation of the MPMC, the ports are assigned as follows: TI_1, TI_2 – first and second clock inputs; Wr_Adr – address-write input; Rd_RAM – RAM-read input; Rd_Adr – address-read input; Wr_RAM – RAM-write input; On – RAM chip-select; Rd_WrInv – RAM read-write-mode select; DB_RAM[15..0] – 16-bit RAM data bus; DB[15..0] – 16-bit data bus; DA_RAM[15..0] – 16-bit RAM address bus. Using the developed components, the schematic of the MPMC was designed, as shown in Fig. 13.

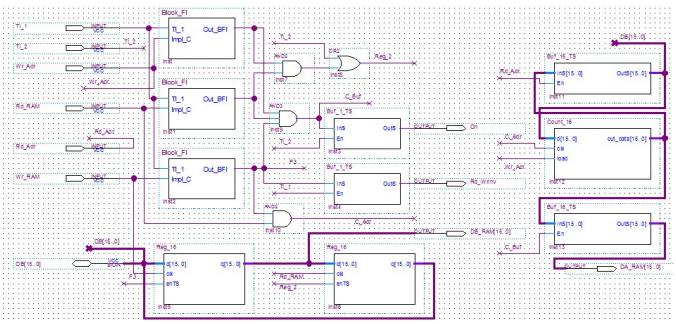


Figure 13. Schematic of the MPMC.

The developed MPMC schematic comprises: input-signal latching and output pulse-generation nodes Block_F_Impl (instances inst, inst1, inst2); Reg_16 registers (instances inst5, inst6) with tri-state outputs; the Count_16 counter (instance inst12); Buf_16_TS 16-bit tri-state buffers outputs (instances inst11, inst13); Buf_1_TS 1-bit tri-state buffers outputs (instances inst3, inst4) and standard logic elements. The timing diagrams of the MPMC in RAM write mode are shown in Fig. 14.

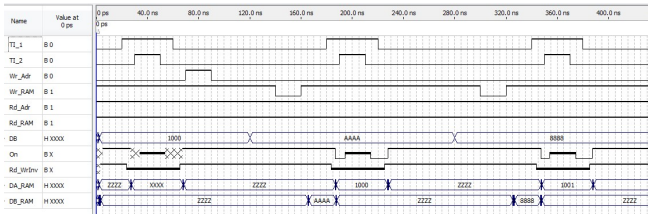


Figure 14. Timing diagrams of the MPMC in RAM write mode

The algorithm for writing data using a MPMC to RAM requires the following steps:

1. On the rising edge of Wr_Adr, data (the address 0x1000) from bus DB is loaded into Count_16.
2. On the rising edge of Wr_Adr, this signal is latched by the pulse-generation block (inst).
3. On the rising edge of TI_1, the pulse-generation block (inst) asserts the signal that releases tri-state buffer (inst13) and connects the counter output to the RAM address bus DA_RAM.
4. On the falling edge of Wr_RAM, data (0xAAAA) from bus DB are loaded into register (inst5).
5. On the falling edge of Wr_RAM, this signal is latched by the pulse-generation block (inst2).
6. On the rising edge of TI_1, the pulse-generation block (inst2) asserts the signal that releases register (inst5) from tri-state and connects its output to the RAM data bus (DB_RAM).
7. Using the output of the pulse-generation block (inst2) together with TI_1, the RAM (Rd_WrInv) read-write signal is formed.
8. On the rising edge of the output of the pulse-generation block (inst2), the counter output increments by one.

The timing diagrams of the MPMC in read mode are shown in Fig. 15.

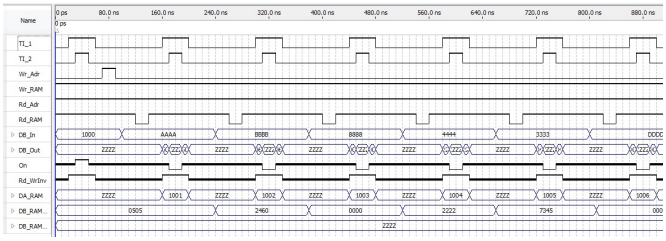


Figure 15. Timing diagrams of the MPMC in RAM read mode

The algorithm for reading data using a MPMC from RAM requires the following steps:

1. On the rising edge of Wr_Adr, data (the address 0x1000) from DB_IN is loaded into Count_16.
2. On the rising edge of Wr_Adr, this signal is latched by the pulse-generation block (inst).
3. On the rising edge of TI_1, the pulse-generation block (inst) asserts the signal that releases tri-state buffer (inst13) and connects the counter output to the RAM address bus(DA_RAM).
4. On the falling edge of Rd_RAM, data (0x0505) from bus DB are loaded into register (inst6).
5. On the falling edge of Reg_2, this signal is latched in the pulse-generation block (inst15).
6. The RAM read-write control (Rd_WrInv) is formed using the output of the pulse-generation block (inst1) together with TI_1.

7. On the rising edge of the output of the pulse-generation block (inst2), the address counter increments by one.

For an implementation of the MPMC with 16-bit address and data buses on an FPGA Cyclone III EP3C16F484C6, the required hardware resources are 114 logic elements and 63 registers.

V. DISCUSSION

Unlike [12], the improved time-division multiple-access method for memory requests reduces access time by pipelining the write (read) process from an external device to RAM, splitting it into two sub-processes: the first preparatory, and the second the direct access to RAM.

The use of bus drivers at the outputs of the MPMC for transmitting addresses, data, and control signals enables the parallel instantiation of the required number of MPMCs. Incorporating three nodes for latching input signals and generating output control signals within the MPMC ensures synchronization of RAM access for asynchronously operating external devices.

The reduction of synthesis time for multichannel, conflict-free data-exchange devices with specified parameters is achieved by using the developed baseline structure (Fig. 4).

In the CGU (Fig. 5), configuring the ROM to generate the sequences of clock pulses TI_{1j} and TI_{2j} for different numbers of external devices, and writing the values m and $3m$ to registers RgAd and RgTI, respectively, provides selection of the specific number of external devices for which the sequences TI_{1j} and TI_{2j} are generated. Adaptation to RAMs with different speeds in the CGU is accomplished by changing the frequency of the input clock pulses applied at TI.

By employing serial-to-parallel and parallel-to-serial converters in the transceiver–MPMC interface adaptation block (Fig. 7), it becomes possible to vary the data-arrival rate at the MPMC input.

Minimization of hardware cost during synthesis of the multichannel conflict-free data-exchange device is achieved by matching the data-arrival intensity P_d (5) with the memory-access intensity P_{MPM} (6).

The use of the proposed basic structure of a multi-channel device for conflict-free data exchange allows to reduce the synthesis time of the required device taking into account the given number of channels. The reduction is achieved by reusing the same type of MPMC, adaptation blocks and CGU, in which the number of channels is given by the parameter m. Quantitative evaluation showed that when switching from the synthesis of an individual solution of a multi-channel device to parametric synthesis based on the proposed basic structure, the development time is reduced, and the relative reduction in synthesis time for devices with 4, 8 and 16 channels is 58.3%, 70.8% and 79.2%, respectively.

It should be noted that the practical implementation of the proposed approach to creating a multi-channel device for conflict-free exchange based on FPGA can also be used in the analysis of large data volumes in image processing systems focused on UAV [24], in visual information classification systems [25], etc.

Despite the results obtained, the proposed model for the synthesis of a multi-channel device for conflict-free data exchange has certain limitations.

First, the model is based on the use of the time division multiple access (TDMA) method, the effectiveness of which depends on the correct choice of the duration of time slots and the number of connected devices. With a significant increase in the number of MRPs, the period of re-access to memory increases, which can lead to an increase in data transfer delays.

Second, the model assumes the use of a centralized storage environment based on multi-port memory. This approach is effective for systems with an average number of exchange channels, but with a large number of MRPs, limitations associated with the bandwidth of the shared memory and data transmission buses may arise.

Third, the work mainly considers the functional aspects of conflict-free data exchange. The issues of ensuring fault tolerance, information protection, memory reservation and adaptive resource redistribution in the event of failures of individual channels have not been investigated and require additional study.

Fourth, the hardware cost estimation was performed for the Cyclone III EP3C16F484C6 FPGA. Therefore, the obtained results require additional verification for modern FPGAs with a higher level of integration and other timing characteristics.

Prospects for further research are related to increasing the scalability and intellectualization of the conflict-free data exchange system. First of all, it is advisable to develop a software-controlled address generator that will provide the formation of complex address sequences for data streaming tasks and reduce memory access time.

An important direction is the creation of algorithms for dynamic allocation of shared memory between mobile robotic platforms that will take into account the current load of communication channels and the priority of information flows. This will allow increasing the efficiency of memory use compared to static resource allocation.

Also promising are the study of mechanisms for adaptive scheduling of TDMA time slots, the use of priority access for critical messages, the integration of cryptographic data protection tools, and the development of fault-tolerant multi-channel architectures for the operation of groups of mobile robotic platforms in real time.

In addition, it is advisable to study the possibilities of implementing the proposed structures on modern FPGAs with further comparison of performance indicators, power consumption, and hardware costs.

VI. CONCLUSIONS

The study accomplishes the stated objectives, specifically.

1. The time-division multiple-access method for arbitrating requests to multi-port memory among external devices was improved by pipelining RAM write/read into two sub-processes (preparatory and execution), reducing memory-access latency to that of a register.
2. A VLSI-oriented structure of the MPMC was devised. By employing input-signal latching and output control-signal generation nodes, an address counter, pipeline registers, and bus drivers, the design enables parallel instantiation of the required number of controllers, a pipelined RAM-access mode, and synchronization of memory access for asynchronously operating external devices.
3. A baseline structure of a multichannel data-exchange

device was developed. Its key components are memory whose capacity depends on the number of connected devices, the sizes of exchanged data arrays, and the control memory required for managing the exchange; the MPMC; the CGU; transceivers; and adapter blocks for the transceiver–MPMC and computer-system–MPMC interfaces. Using this baseline reduces the synthesis time of a parameterized multichannel device.

4. A CGU structure was proposed which, using ROM, input clock pulses *TI*, a counter *LichTI*, and registers Rg*Ad* and Rg*TI*, supports configuration to the RAM throughput and device count by generating the clock-pulse sequences TI_{1j} and TI_2.
5. An interface-adaptation block for the transceiver–MPMC boundary was designed. By employing serial-to-parallel and parallel-to-serial conversions, it allows adjustment of the data-arrival rate at the MPMC input.
6. A synthesis method for a multichannel conflict-free data-exchange device was formulated. It accounts for application-specific requirements, uses the baseline device structure together with the MPMC, adaptation blocks, and CGU, and reduces hardware cost by matching data-arrival intensity to memory-access intensity.
7. The simulation of the CPU was performed in the Quartus development environment. It was found that the implementation of the CPU with 16-bit address and data buses on the Cyclone III EP3C16F484C6 FPGA requires 114 logic elements and 63 registers. It is shown that the pipelined memory access process reduces the cycle time of accessing the CPU, which provides an increase in system throughput compared to the traditional access scheme. The use of the basic parametric structure provides a reduction in the synthesis time of a multi-channel device by 58–79%, and matching the intensity of data flows with the memory bandwidth allows minimizing hardware costs.

References

[1] I.G. Tsmots, Yu.V. Opotiak, K.M. Obelovska, S.V. Tesliuk, "Methods and means of conflict-free data exchange in the group of mobile robotic platforms," *Ukrainian Journal of Information Technology*, vol. 6, no. 1, pp. 65–75, 2024. https://doi.org/10.23939/ujit2024.01.065.

[2] I. Tsmots, V. Teslyuk, A. Łukaszewicz, Y. Lukashchuk, I. Kazymyra, A. Holovatyy, Y. Opotyak, "An approach to the implementation of a neural network for cryptographic protection of data transmission at UAV," *Drones*, vol. 7, p. 507, 2023. https://doi.org/10.3390/drones7080507.

[3] D.S. Ilcev, "Analyses of time division multiple access (TDMA) schemes for global mobile satellite communications (GMSC)," *TransNav*, vol. 14, no. 4, pp. 831–837, 2020. https://doi.org/10.12716/1001.14.04.06.

[4] S.D. Ilcev, "Analyses of space division multiple access (SDMA) schemes for global mobile satellite communications (GMSC)," *TransNav*, vol. 14, no. 4, 2020. https://doi.org/10.12716/1001.14.04.05.

[5] S.D. Ilcev, "Analyses of code division multiple access (CDMA) schemes for global mobile satellite communications (GMSC)," *TransNav*, vol. 14, no. 4, 2020. https://doi.org/10.12716/1001.14.04.03.

[6] *Telecomunicazioni*, "TDMA, FDMA, and CDMA," University of Rome La Sapienza, Rome, Italy, 2008, 42 p.

[7] G. Garvey, *Multiple Access Techniques*, Atlanta RF, LLC, Roswell, GA, 2013, 53 p.

[8] X. Xu *et al.*, "Combination of asymmetric PCMA and common multiple access technology," in *Proc. Int. Conf. (TLICSC)*, 2018, pp. 1–9. https://doi.org/10.2991/tlicsc-18.2018.36.

[9] W. Wolf, *Modern VLSI Design: IP-Based Design*, 4th ed., Upper Saddle River, NJ: Prentice Hall, 2006.

[10] J.-L. Lin, B.-C.C. Lai, "BRAM-efficient multiported memory on FPGA" *Proc. Int. Symp. VLSI Design, Automation and Test (VLSI-DAT)*, Apr. 2015, pp. 1–4. https://doi.org/10.1109/VLSI-DAT.2015.7114526.

[11] A. Pandey, B. Suhas A, B.P. and S. Madhavan, "Design and implementation of synchronous dual-port memory," *Proceedings of the 2024 Third Int. Conf. on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, Ballari, India, 2024, pp. 1–6. https://doi.org/10.1109/ICDCECE60827.2024.10549556.

[12] X.-T. Nguyen, D.-H. Le, T.-T. Bui, H.-T. Huynh, C.-K. Pham, "A flexible high-bandwidth low-latency multi-port memory controllers," *Vietnam Journal of Science and Technology*, vol. 56, no. 3, pp. 357–369, 2018. https://doi.org/10.15625/2525-2518/56/3/11103.

[13] T. Subhashini, M. Kamaraju and K. Babulu, "Design and analysis of multiple port memory architecture for low power applications," *Proceedings of the 2018 Conference on Signal Processing and Communication Engineering Systems (SPACES)*, Vijayawada, India, 2018, pp. 210-214. https://doi.org/10.1109/SPACES.2018.8316348.

[14] K.S. Sikand, L. Zartman, S. Rabiee, J. Biswas, "Robofleet: Open source communication and management for fleets of autonomous robots," *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Prague, Czech Republic, 2021, pp. 406–412. https://doi.org/10.1109/IROS51168.2021.9635830.

[15] K. Sethi, "Design space exploration of algorithmic multi-port memories in high-performance application-specific accelerators," arXiv preprint arXiv:2007.09363, 2020. [Online]. Available at: https://doi.org/10.48550/arXiv.2007.09363.

[16] M. Vegni, V. Loscrí, C.T. Calafate, P. Manzoni, "Communication technologies enabling effective UAV networks: A standards perspective," *IEEE Communications Standards Magazine*, vol. 5, no. 4, pp. 33–40, 2021. https://doi.org/10.1109/MCOMSTD.0001.2000074.

[17] E. Pereira, L. Luza, N. Moura, L. Ost, N. Calazans, F. Moraes, R. Garibotti, "Assessment of communication protocols' latency in co-processing robotic systems," *Proceedings of the 2023 21st IEEE Interregional NEWCAS Conference (NEWCAS)*, 2023, pp. 1–5. https://doi.org/10.1109/NEWCAS57931.2023.10198085.

[18] R. Siegwart, I.R. Nourbakhsh, D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, MIT Press, 2011, 472 p.

[19] X. Wang, C. Yuen, S.H. Dau, "Delay minimization for network coded cooperative data exchange with rate adaptation," *Proceedings of the 2013 IEEE 78th Vehicular Technology Conference (VTC Fall)*, Las Vegas, NV, USA, 2013, pp. 1–5. https://doi.org/10.1109/VTCFall.2013.6692154.

[20] N. Koul, N. Kumar, A. Sayeed, C. Verma, M.S. Raboaca, "Data exchange techniques for Internet of Robotic Things: Recent developments," *IEEE Access*, vol. 10, pp. 102087–102106, 2022. https://doi.org/10.1109/ACCESS.2022.3209376.

[21] I.I. Ivanov, Element base of electronic device, Kyiv: Tekhnika, 2020, 320 p. (in Ukrainian).

[22] W. Wolf, *FPGA-Based System Design*, Upper Saddle River, NJ: Prentice Hall, 2004.

[23] B.A. Demyda, Yu.M. Rashkevych, I.H. Tsmots, Multi-port memory, Patent of Ukraine No. 23358, 1998, Bulletin no. 4. (in Ukrainian).

[24] Y. Zheng, et al., "Wavelet transform cluster analysis of UAV images for sustainable development of smart regions due to inspecting transport infrastructure," *Sunstainability*, vol. 17, issue 3, 927, 2025. https://doi.org/10.3390/su17030927.

[25] W. Huan, et al., "Haar wavelet-based classification method for visual information processing systems," *Applied Sciences*, vol. 13, issue 9, 5515, 2023. https://doi.org/10.3390/app13095515.

Ivan Tsmots received the degree of Doctor of Technical Sciences in 2007. He is currently a Professor at the Department of Automated Control Systems, Lviv Polytechnic National University. His research areas include multi-port memory controllers and conflict-free data exchange, neural and neuro-fuzzy hardware elements, and architectures for mobile robotic platforms.

Sofiia Tesliuk graduated from Lviv Polytechnic National University with a Master's degree with honors in 2022. Currently working as a Postgraduate student at the Department of Automated Control Systems, Lviv Polytechnic National University. Her research areas include conflict-free data exchange for groups of mobile robotic platforms, multi-port memory controllers, and automated network-traffic analysis.

Yurii Opotyak received the degree of Candidate of Technical Sciences (PhD equivalent) in 1997. He is currently a Senior Researcher and Associate Professor at the Department of Automated Control Systems, Lviv Polytechnic National University. His research areas include models and tools for debugging and testing mobile systems, IoT and telecommunications, neuro-like cryptographic protection of data transmission, and architectures for mobile robotic platforms with intelligent motion control.

Volodymyr Antoniv received the philosophy of doctorate degree in Information Technologies in 2021. He is currently working as a Senior Lecturer at the Department of Automated Control Systems, Lviv Polytechnic National University. His research areas include neuro-fuzzy control for groups of mobile robotic platforms, means and methods for data processing, and architectures for autonomous and robotic systems.